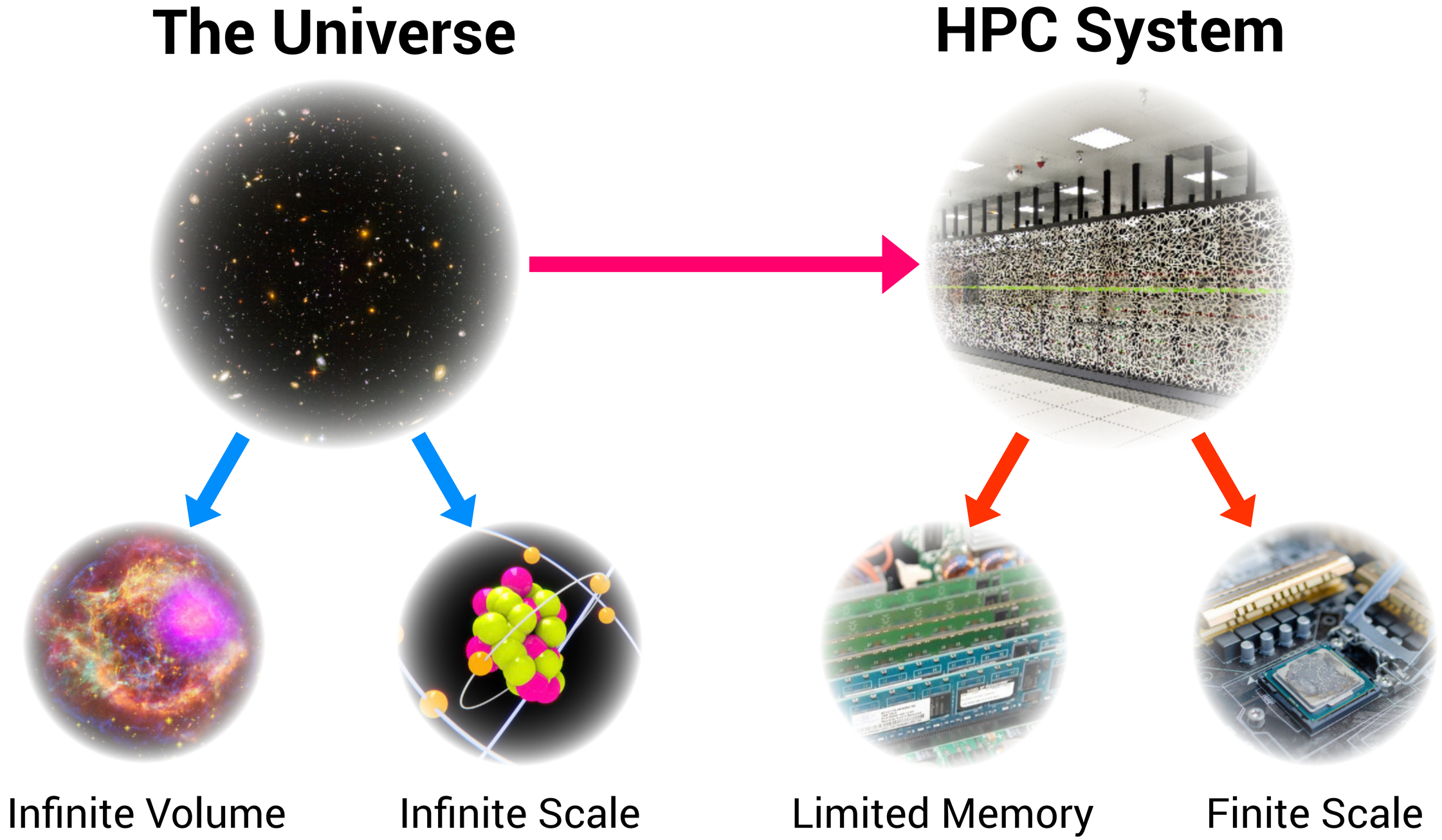
Simulating the Universe at Petascale

Josh Borrow
Institute for Computational Cosmology, Durham University

DiRAC intel SWIFT Durham University ICC

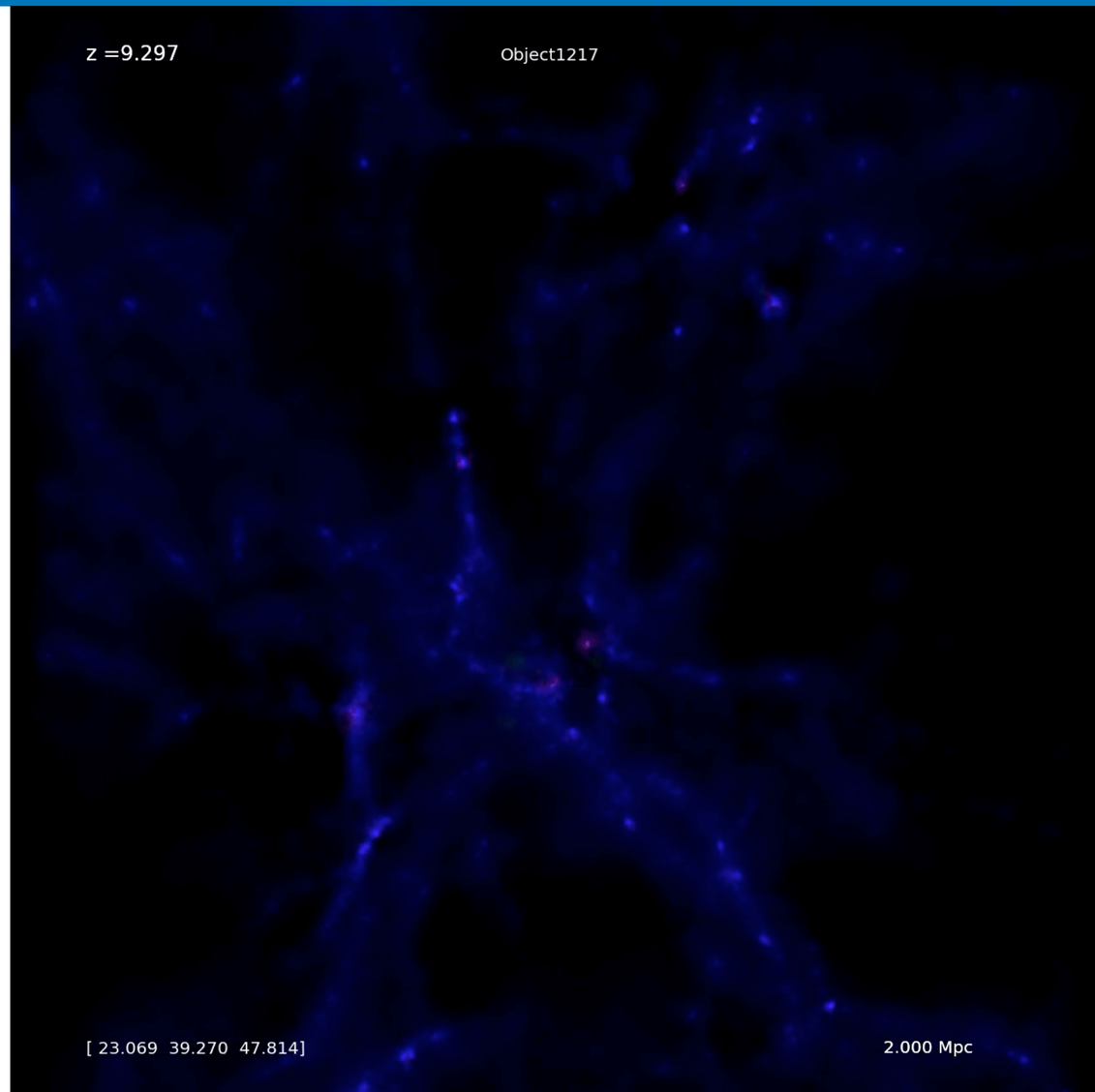

The Universe
HPC System
Infinite Volume
Infinite Scale
Limited Memory
Finite Scale

To what end?

- Astronomers only see a snapshot of the universe
- Impossible to see time-variation
- Linking **properties of galaxies with their dynamical evolution** and environment is then very difficult
- Simulations allow us to do this.

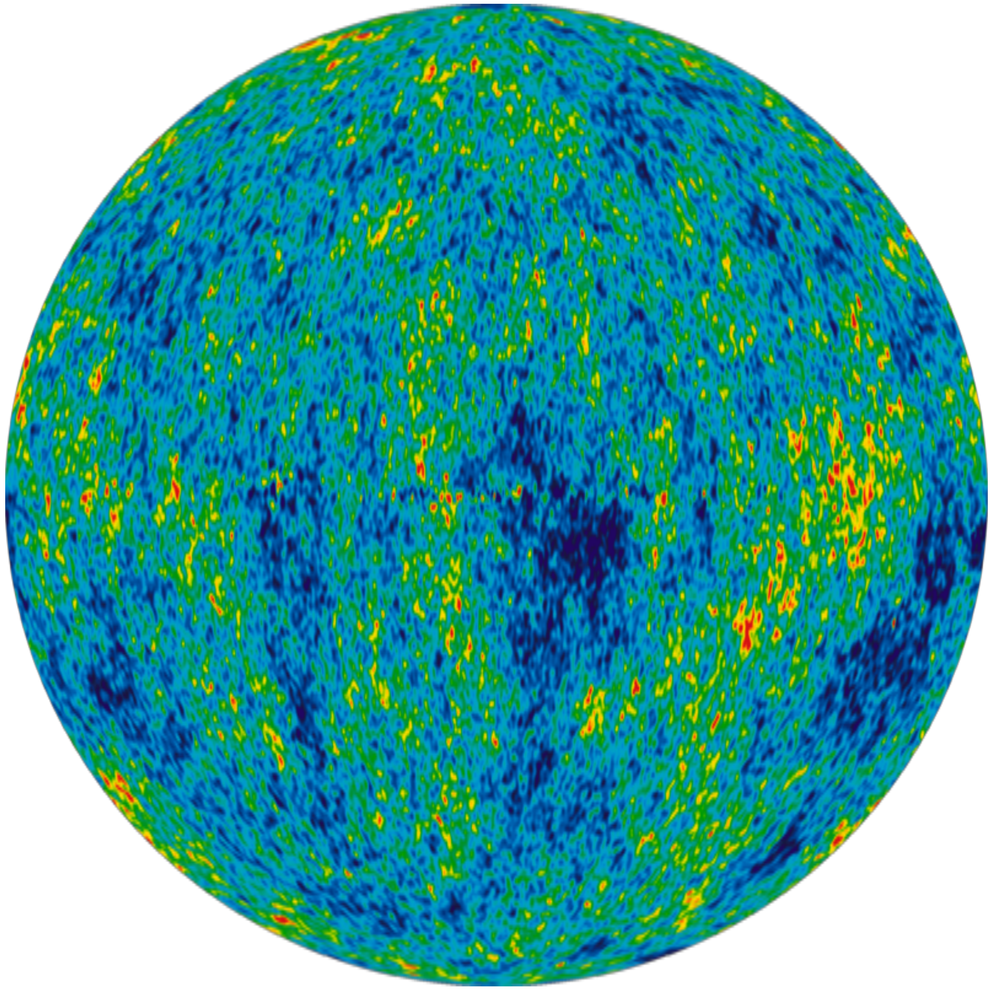
To what end?

- Astronomers only see a snapshot of the universe
- Impossible to see time-variation
- Linking **properties of galaxies with their dynamical evolution** and environment is then very difficult
- Simulations allow us to do this.



Running a simulation

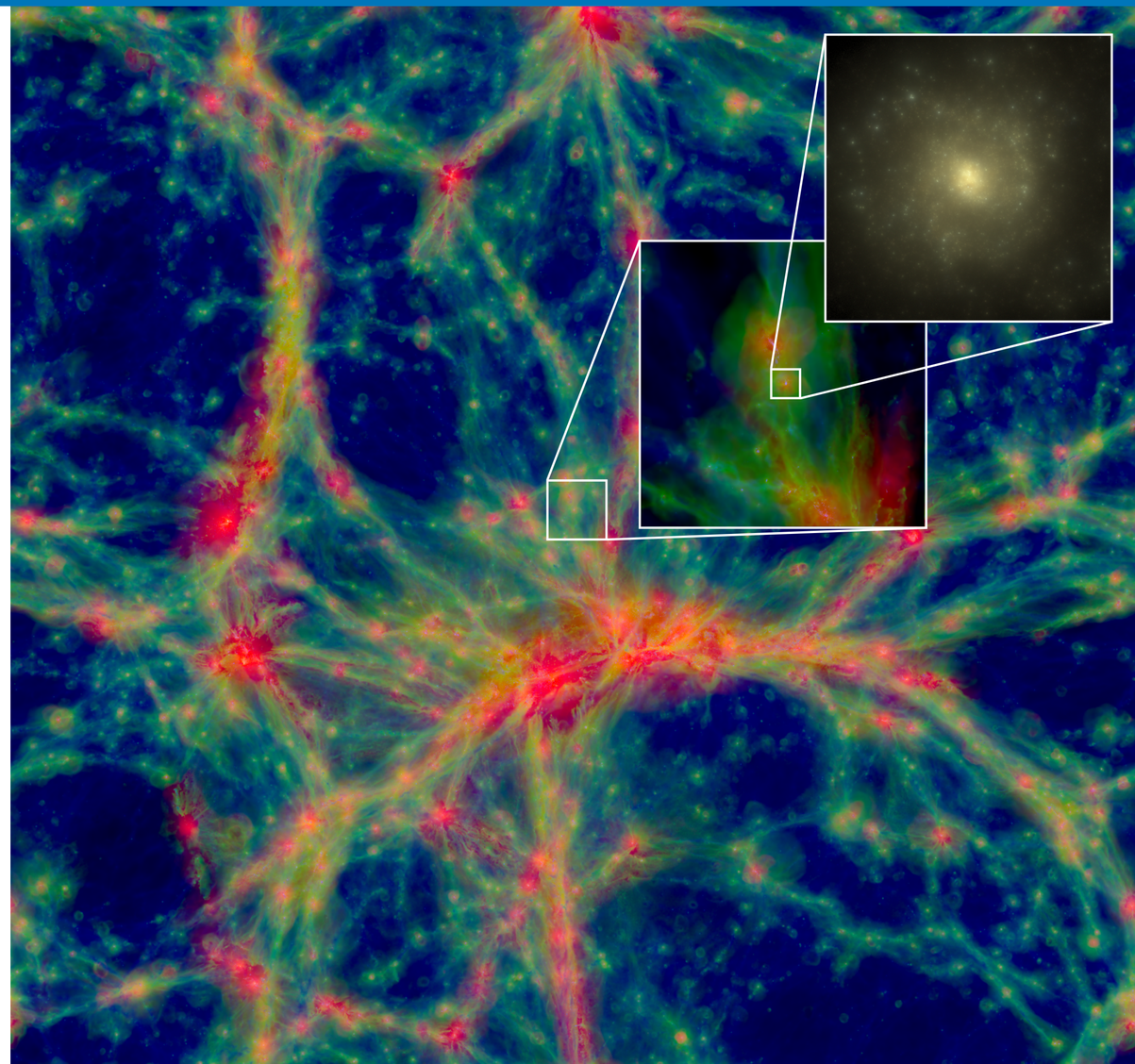
- Take a big bit of Universe (1 billion lightyears on a side)
- Get your **boundary conditions** from the Cosmic Microwave Background (380’000 years after big bang)
- Press play
- …
- Take a look at the galaxies you get



The cosmic microwave background (CMB) as seen by WMAP

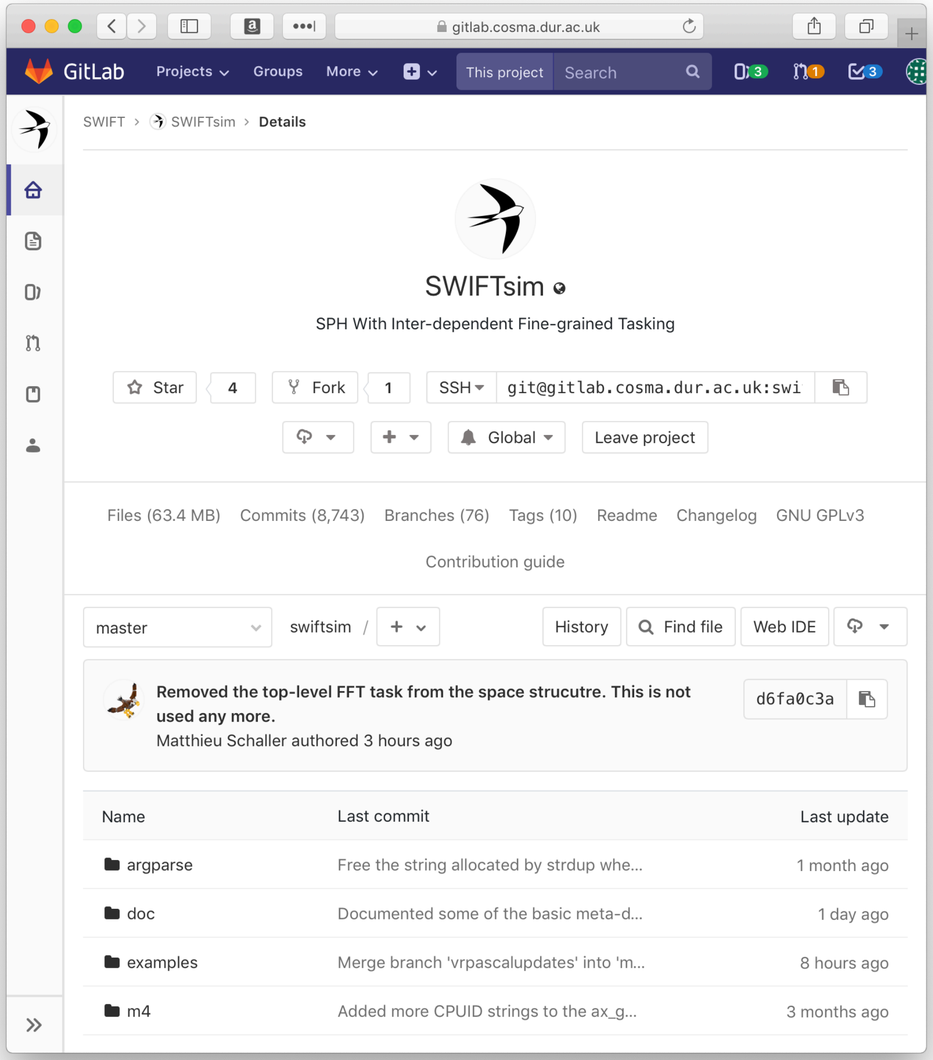
On what scale?

- Our next big run:
- Nearly **100 billion particles**
- 30’000 galaxies with a mass similar to our own Milky Way
- Each galaxy resolved by 10’000 - 100’000 particles



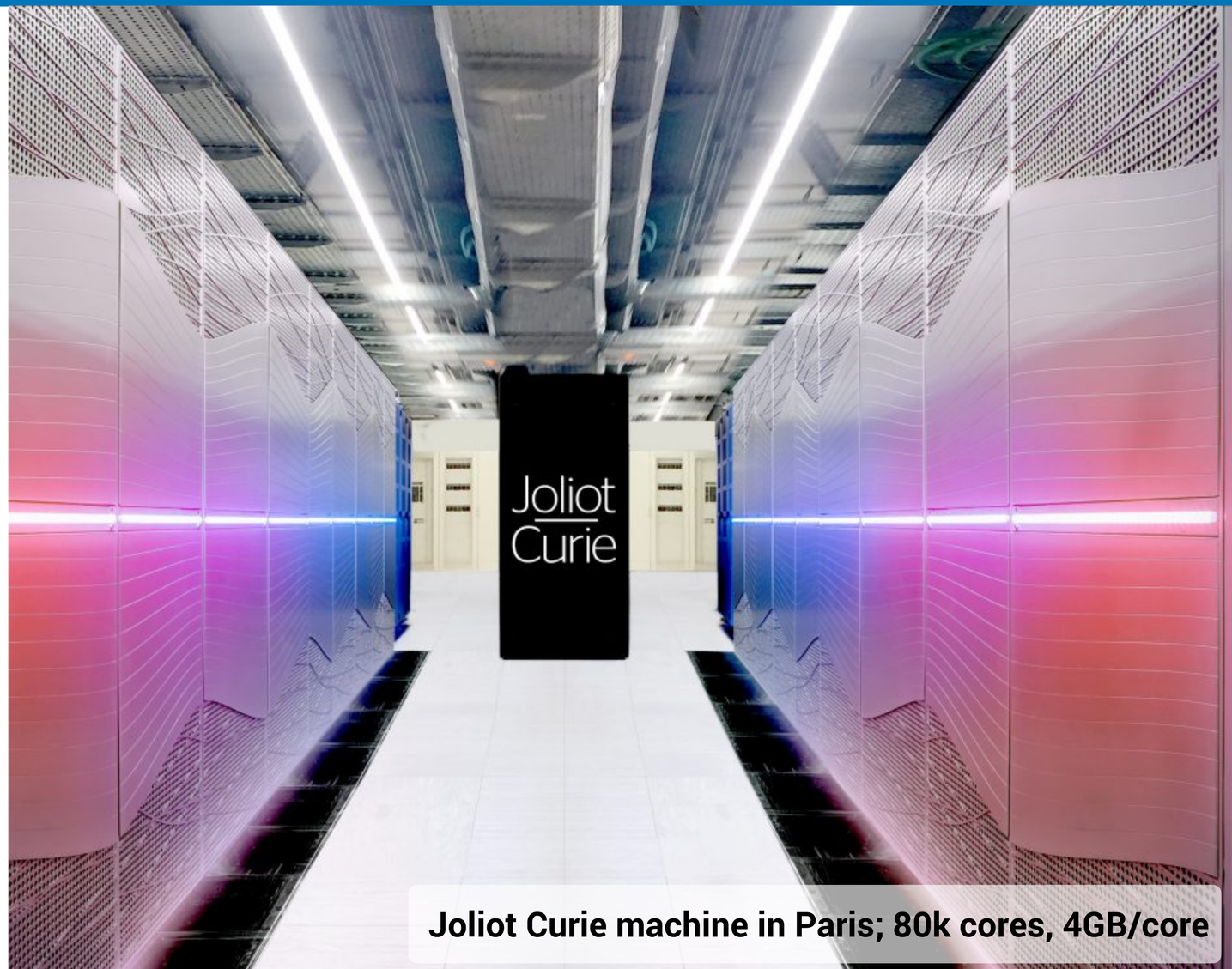
What kind of code?

- 100’000 lines of open-source C99 code
- **Hybrid** (MPI + threads)
- Collaboration between astronomy, computer science, and industry
- Supported by DiRAC



What kind of resources?

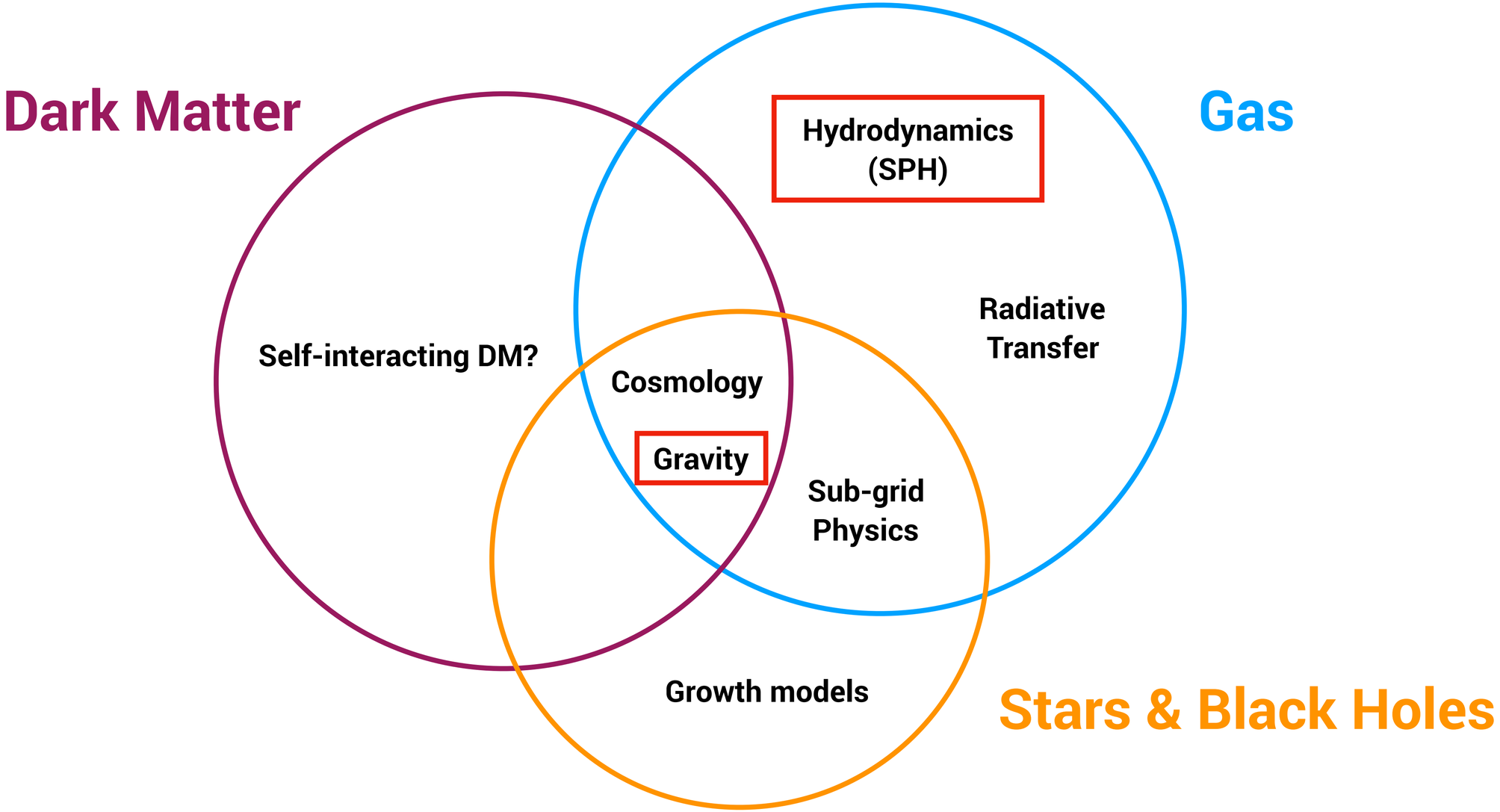
- Need Tier-0 resources
- 10s of millions of core hours
- 10’000s of CPUs
- **~200-300 TB of RAM**
- Snapshots ~4-5 TB each



Joliot Curie machine in Paris; 80k cores, 4GB/core

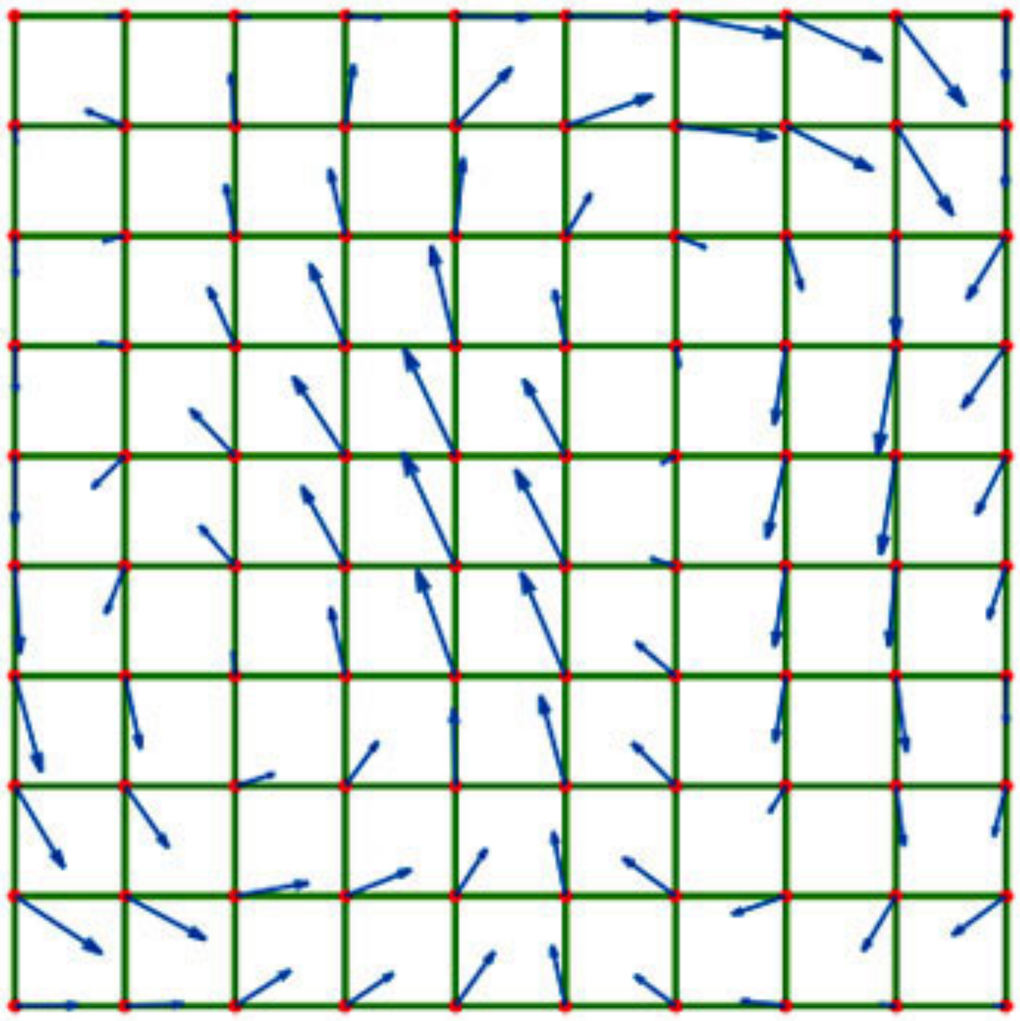
Techniques

What do we need?

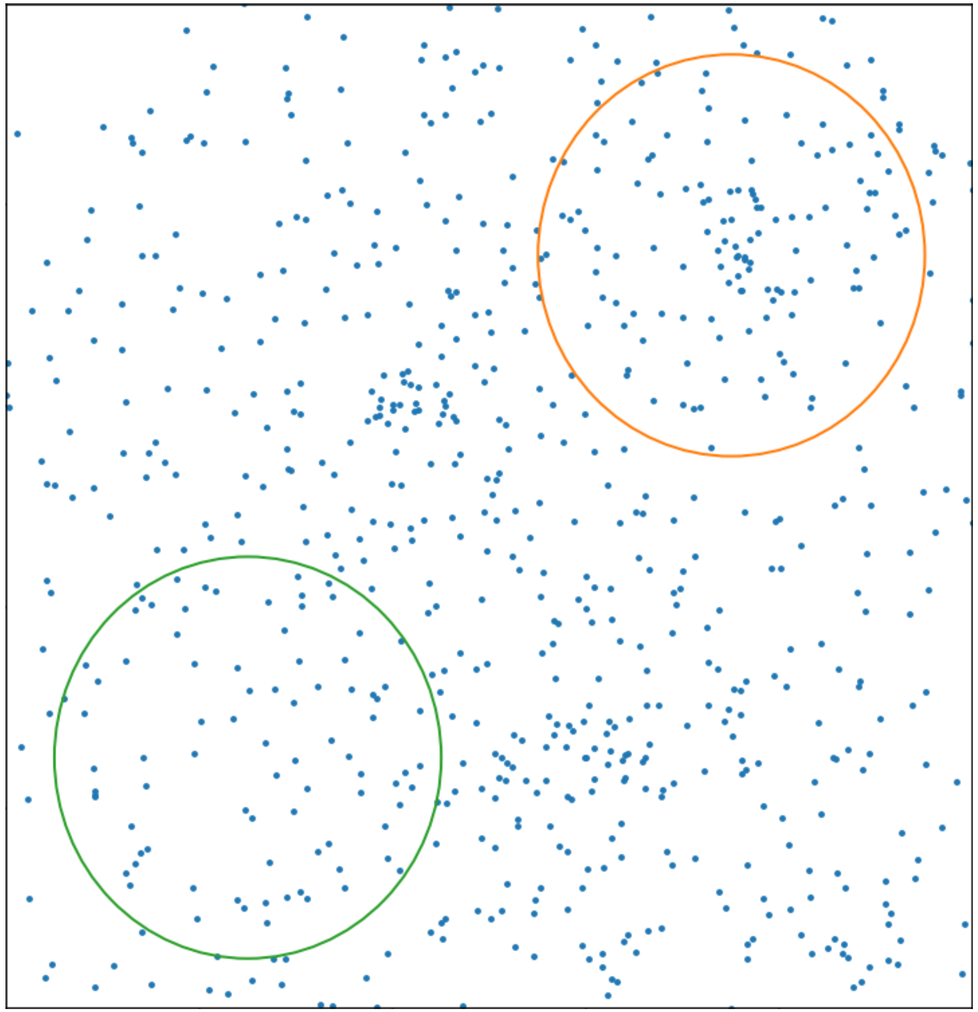


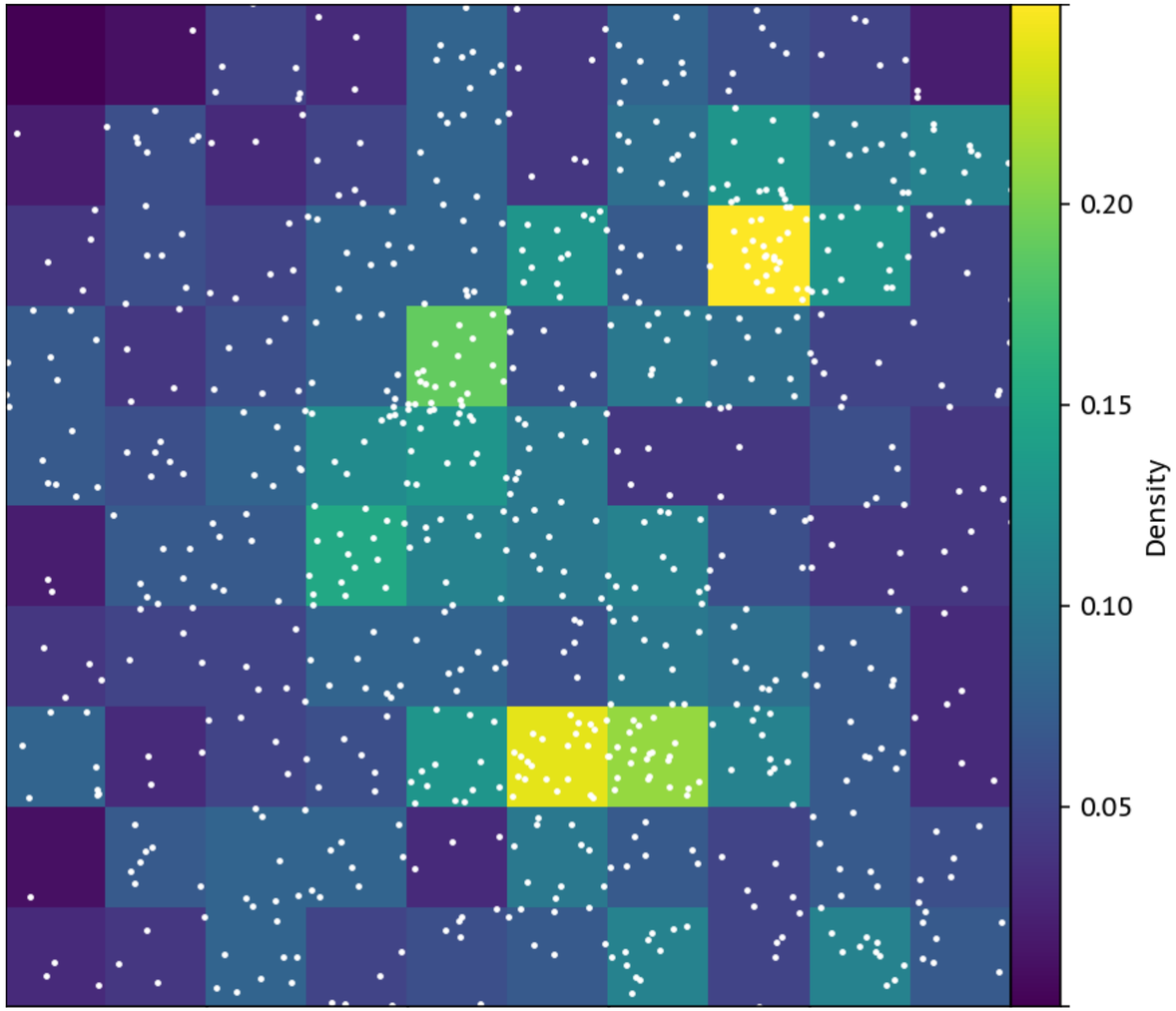
Hydrodynamics

- Typical way to do this: lay down a grid and compute fluid flows between cells
- Several problems:
  - Lack of adaptivity
  - **Cannot track fluid flow over time**
  - Makes gravity difficult



Smoothed Particle Hydrodynamics (SPH)

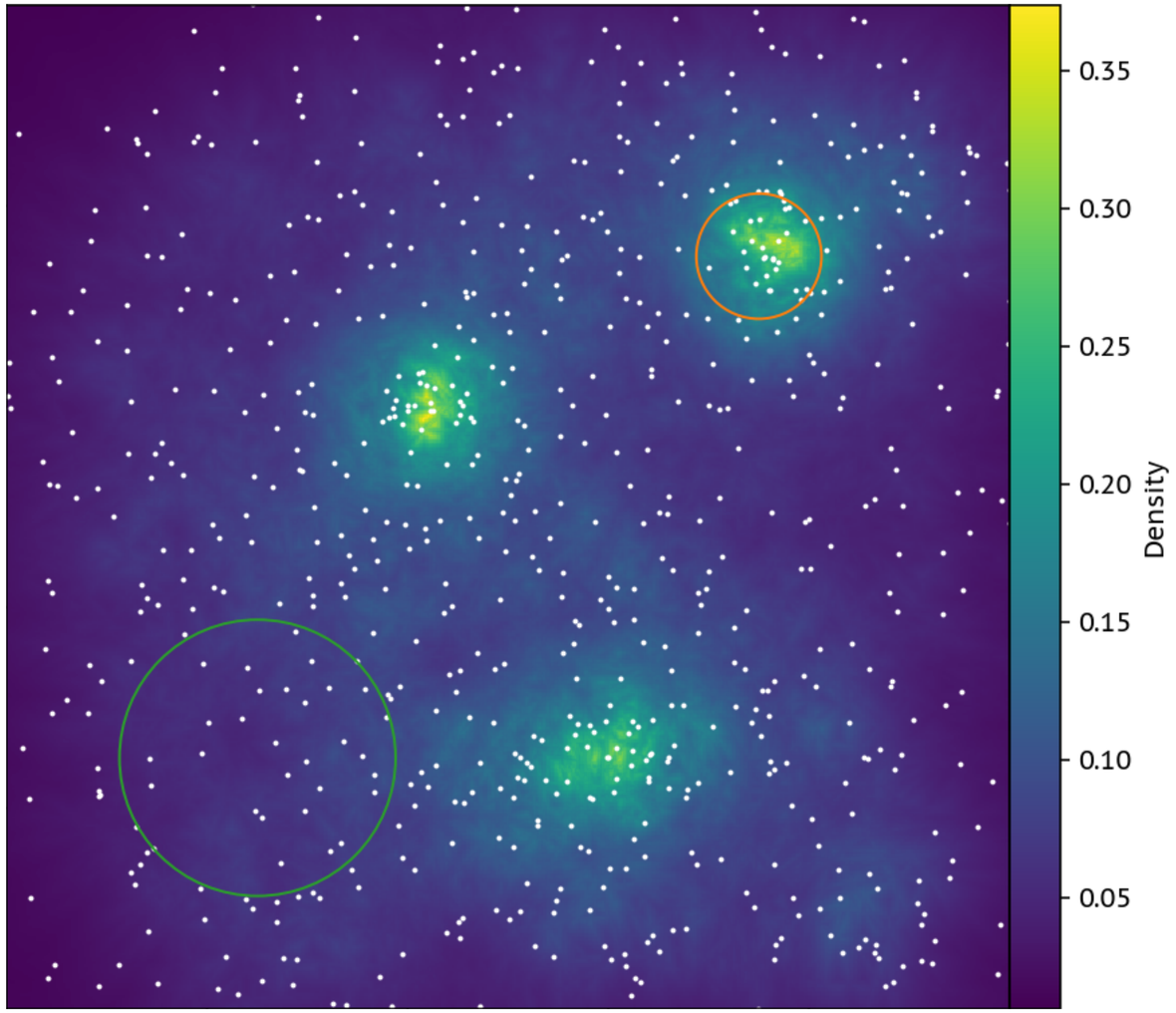




Improving our density estimate

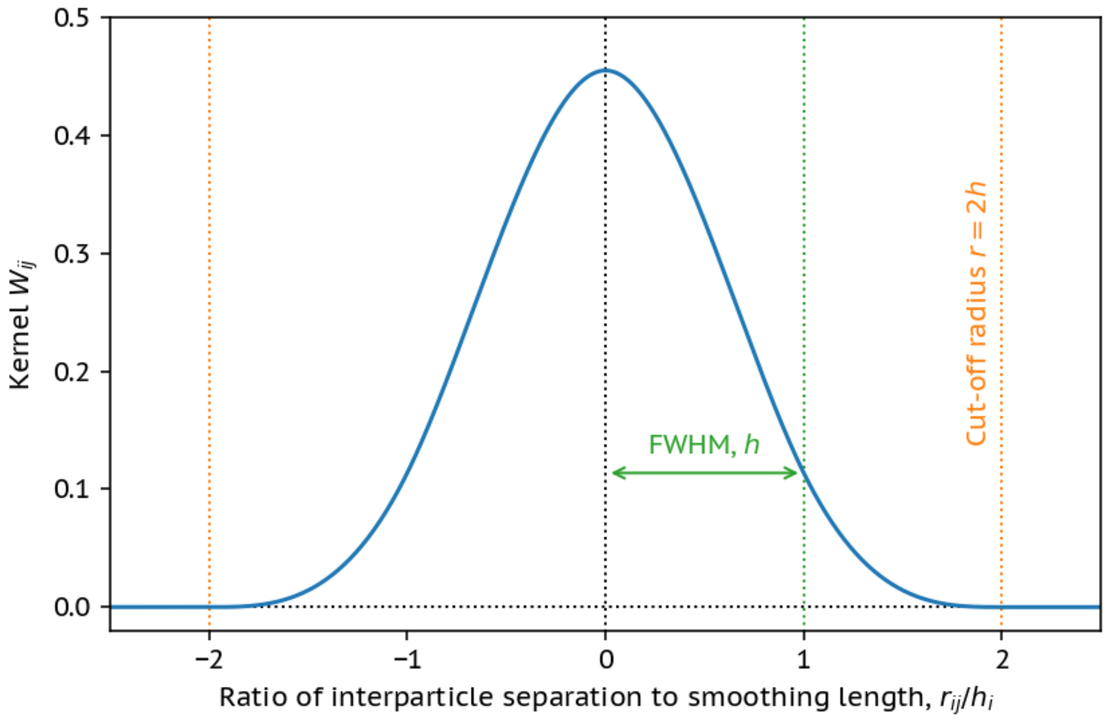
- What if we **change the volume** we consider for each particle?
- Set *R* such that our volume encloses 30 particles.
- Density for each particle is then

$$\rho_i = \frac{30 m_{\mathrm{part}}}{\frac{4}{3}\pi R_i^3}$$



Enter Gaussian filtering?

- We can do better! We **care less about particles that are further away**.
- Weight the contribution from each particle with a “kernel”.
- Size kernel to enclose 30 particles within cut-off radius.

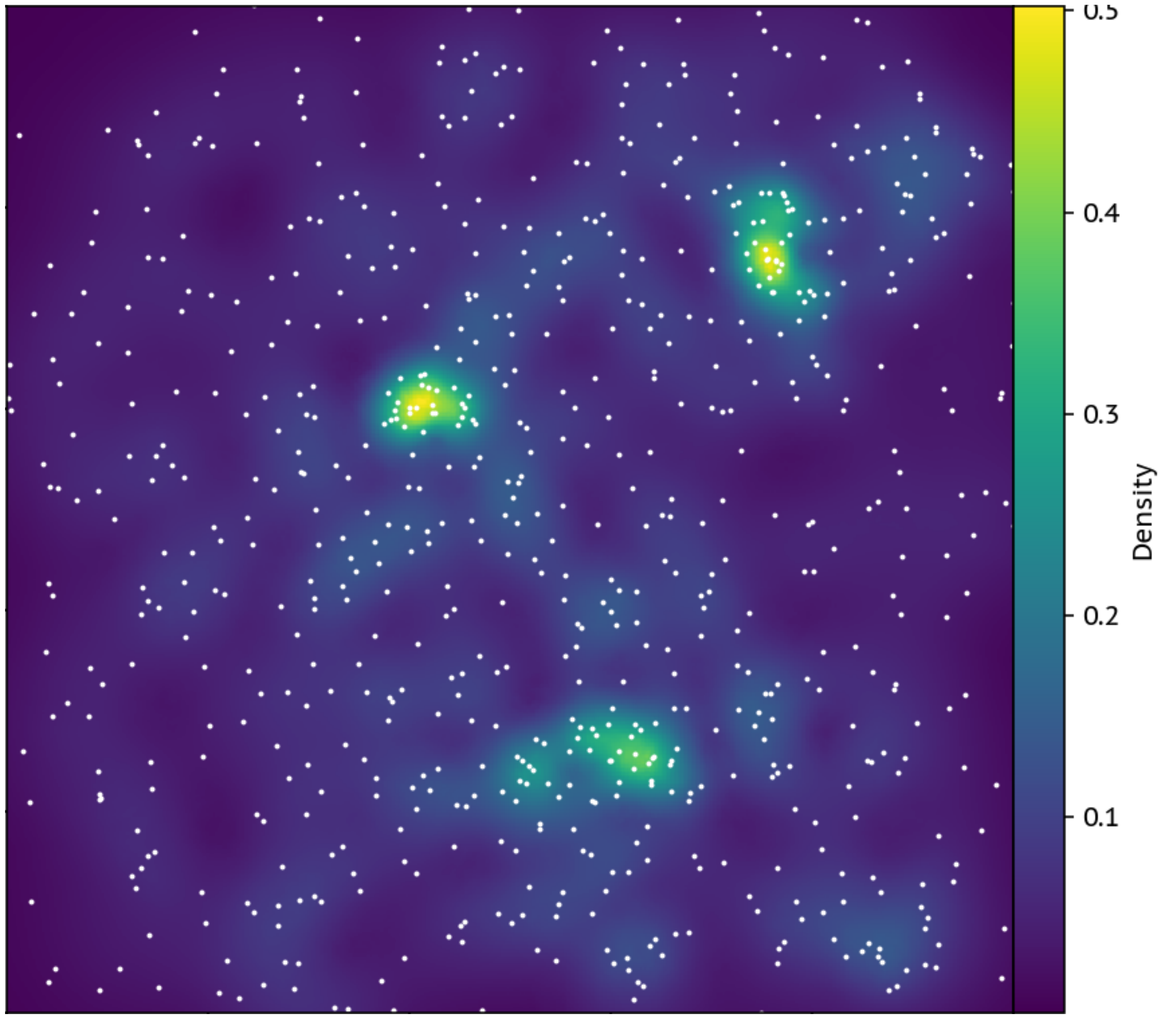


A smooth density estimate

- Density for each particle now given by:

$$\rho_i = \sum_{j=1}^{30} m_{\mathrm{part}} W(r_{ij}, h_i)$$

- Density (+ temperature, tracked by particles) gives **pressure, which gives dynamics**



Adding in the ‘Dynamics’

$$L(q,\dot{q}) = \frac{1}{2}\sum_{i=1}^{N} m_i \dot{r}_i^2 - \sum_{i=1}^{N} m_i u_i,$$

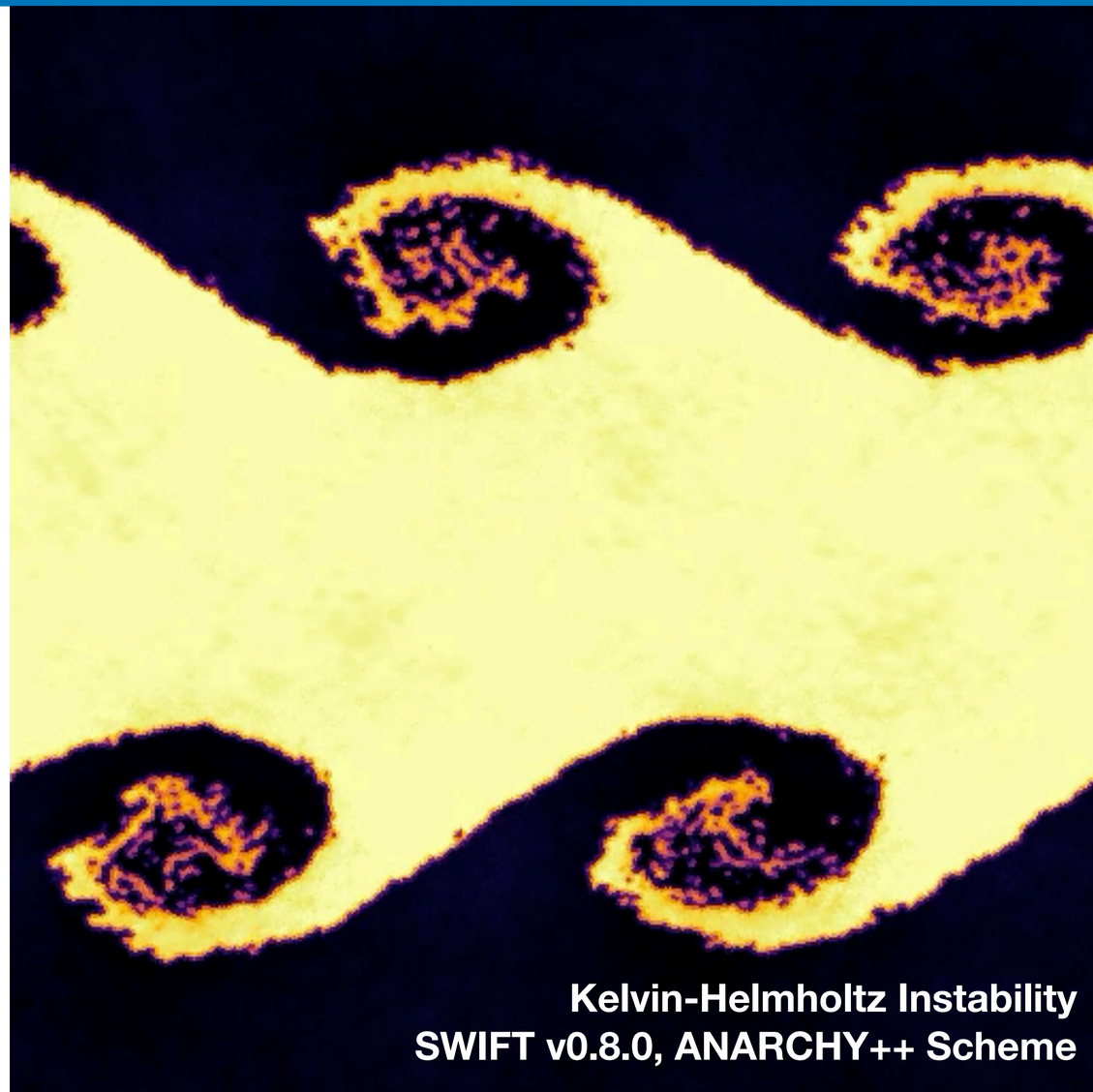
$$\left.\frac{\partial u_i}{\partial q_i}\right|_A = -\frac{P_i}{m_i}\frac{\partial \Delta V_i}{\partial q_i},$$

+

$$\phi_i(\mathbf{q}) = \kappa h_i^{n_d}\frac{1}{\Delta \tilde{V}} - N_{ngb} = 0,$$

$$m_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \sum_{j=1}^{N} P_j \nabla_i \Delta \tilde{V}_j \psi_j + P_j \nabla_i \Delta V_j.$$

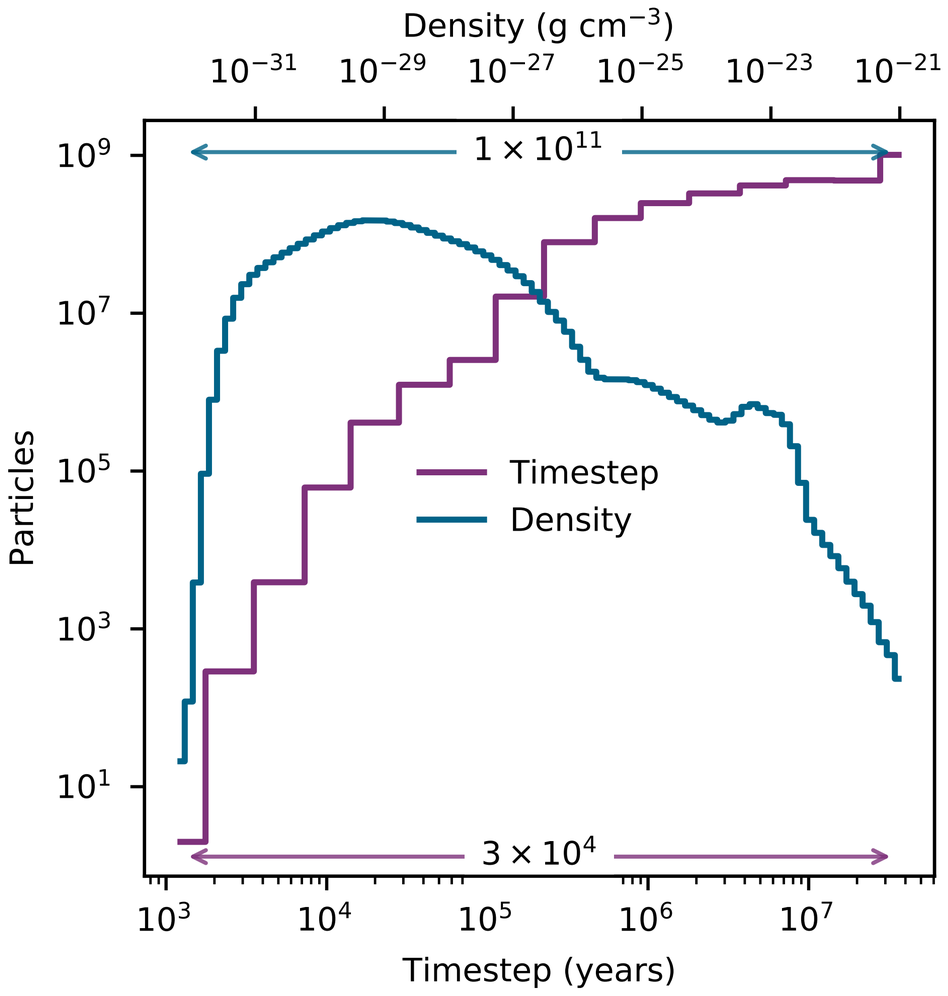
$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} x_i x_j \left[\frac{f_{ij}P_i}{y_i^2}\nabla_i W_{ij}(h_i) + \frac{f_{ji}P_j}{y_j^2}\nabla_i W_{ji}(h_j)\right]$$



Simulating at Scale

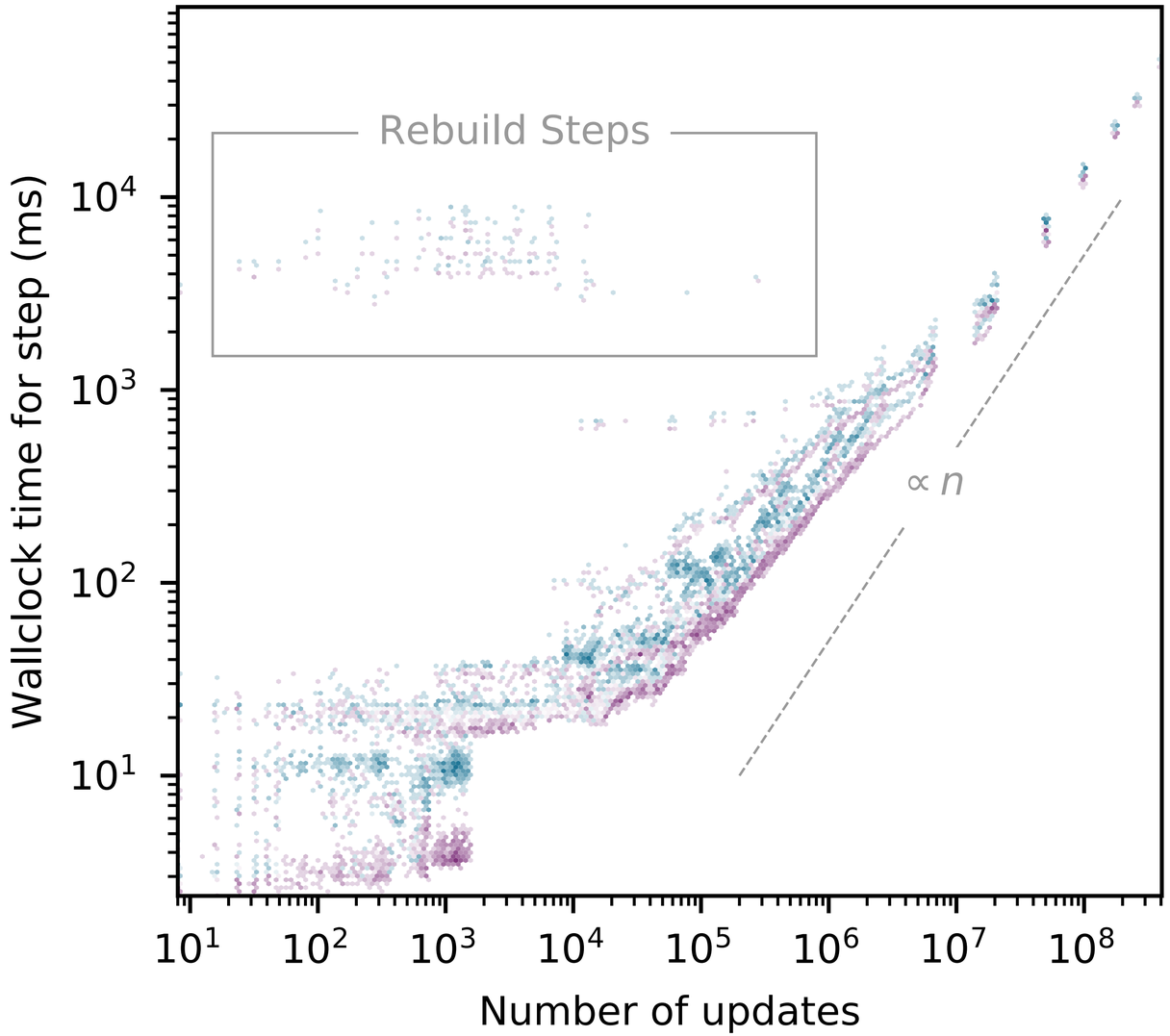
Why is hydrodynamics so challenging?

- We must use a multiple time-stepping scheme; each particle has it's own *dt*
- This is required because of the **huge range in density** in these simulations (voids through to the cores of galaxies)
- Not using such a scheme causes a ~**100-1000x** degradation in performance.



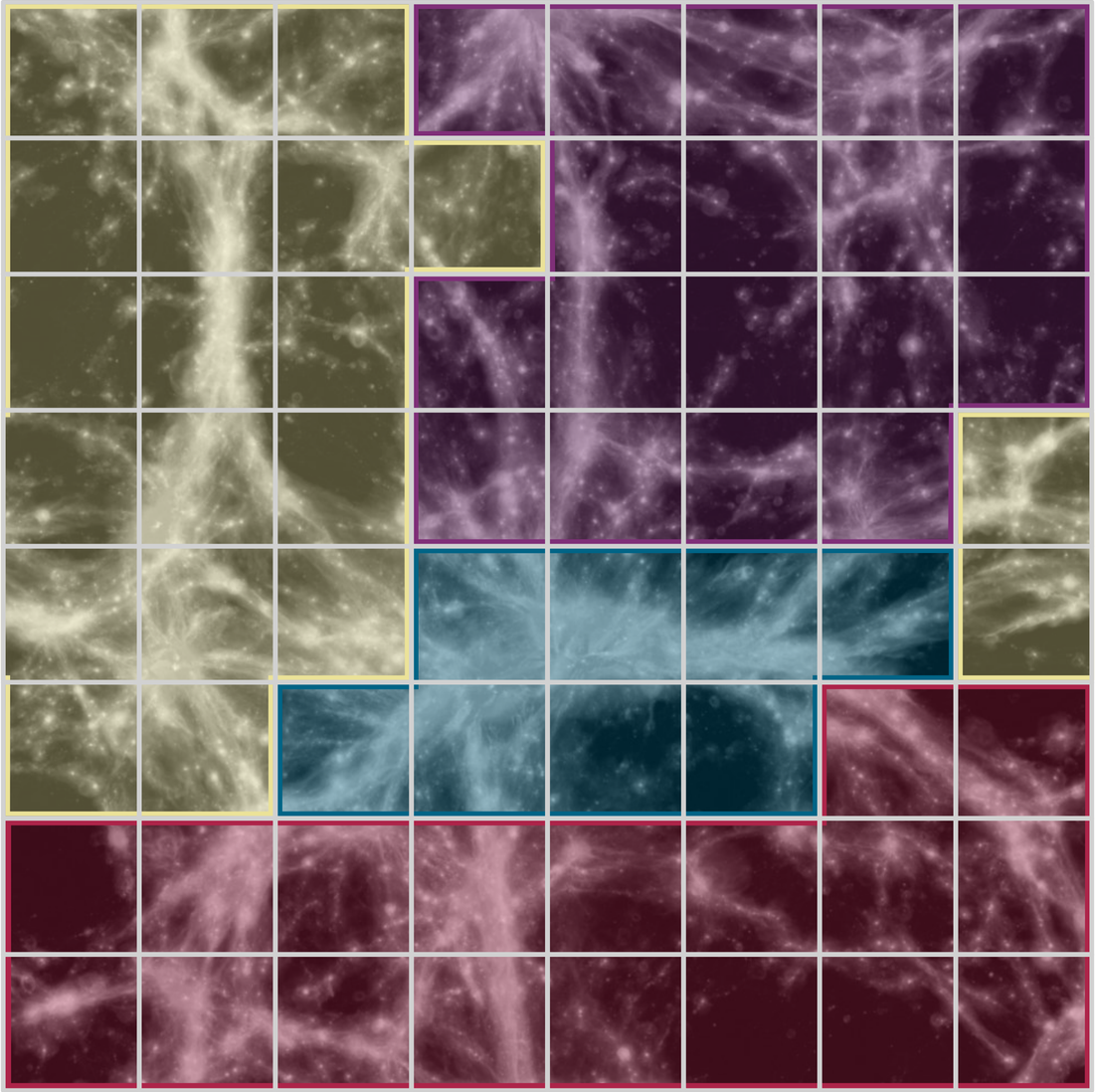
How does this cause a problem?

- Each dot represents a step in the simulation
- “easy” to scale large steps
- Small steps very difficult - **handful of particles** (out of 200TB worth of ram full) **active**
- Impossible to load balance



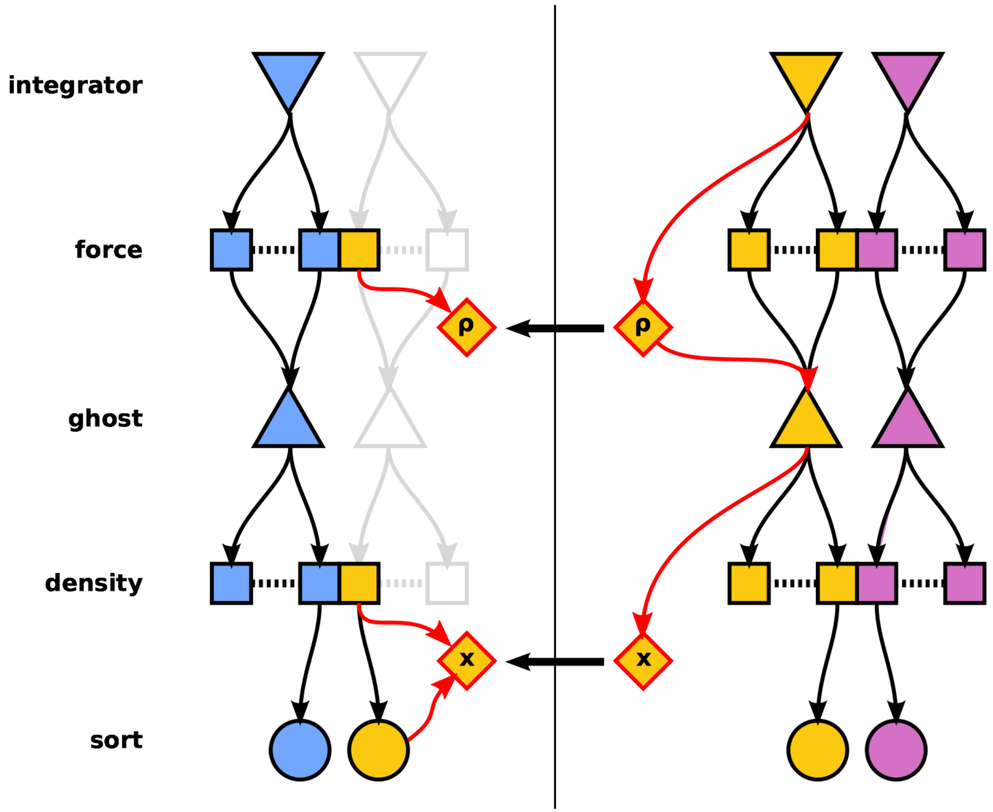
Damage control

- Ensure there is **no MPI** (communication) **in the short steps**
- Do this by using a strategy that places the particles for a galaxy at the “center” of a node
- This ensures short steps are as short as possible; latency kills you.



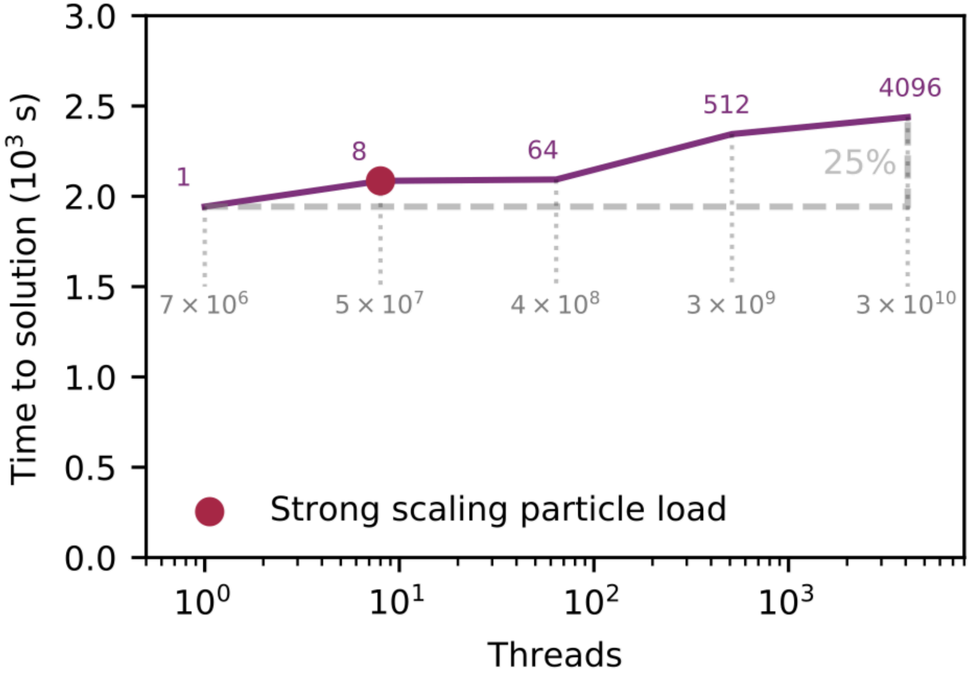
Other incremental improvements

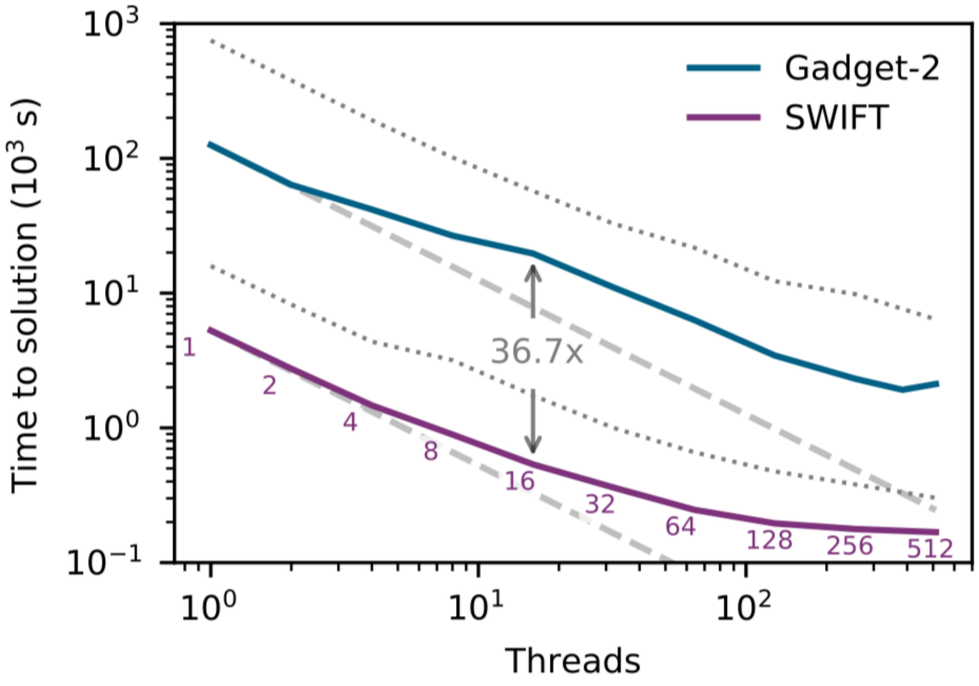
- Hand-written AVX-512 kernels for particle interactions (~3-4x speedup)
- Smart algorithms for finding neighbours (~5x speedup)
- Asynchronous communications & better load balancing (~2x speedup)



Does this actually work?

- Yes!
- Normally weak-scaling such a problem is incredibly difficult.
- Larger simulations -> more dense regions -> larger time-step hierarchy.

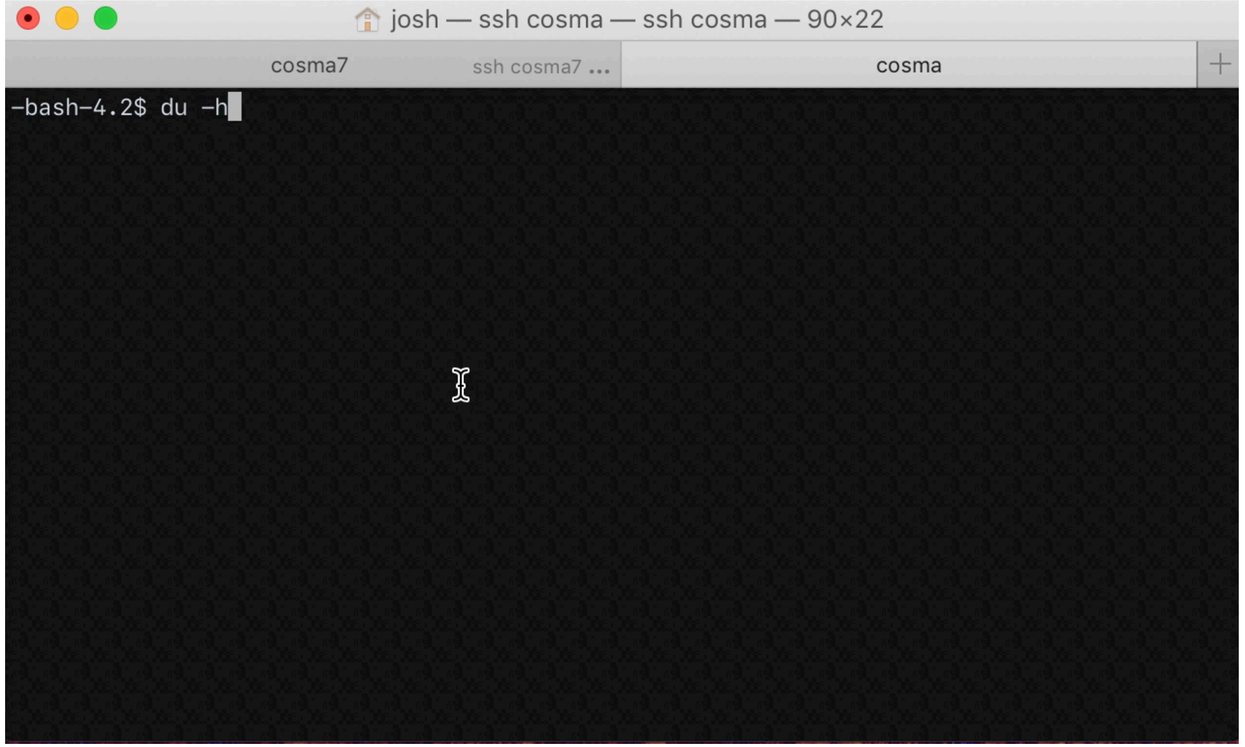




Dealing with Data

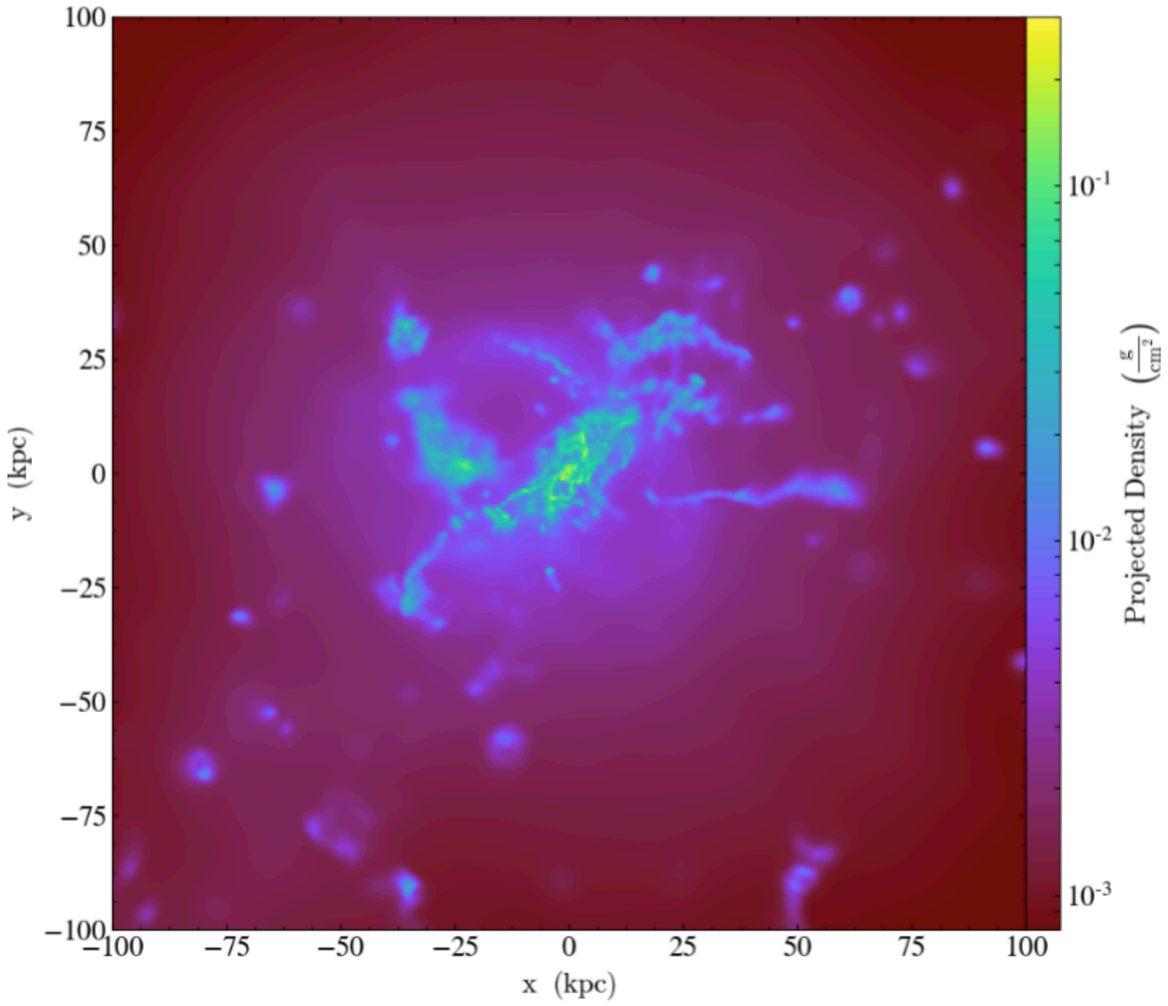
Data sizes

- A full run of our production simulation will produce **3 PB** of data.
- This is untenable to store!
- Need on-the-fly analysis



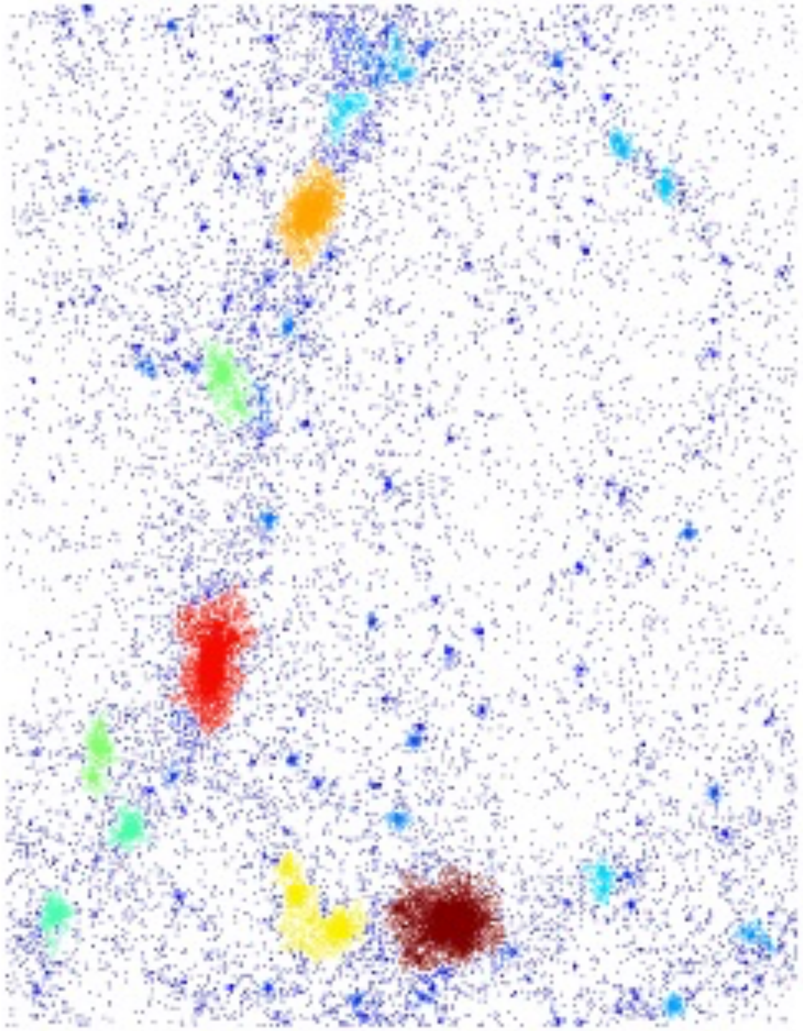
Choices of tools

- Unfortunately, `pandas.df("cosmological_simulation")` doesn't quite cut it
- Pandas has a 5-10x memory overhead and cannot deal well with partial reads
- We have **custom-built tools** (e.g. yt & unyt NUMFOCUS projects)



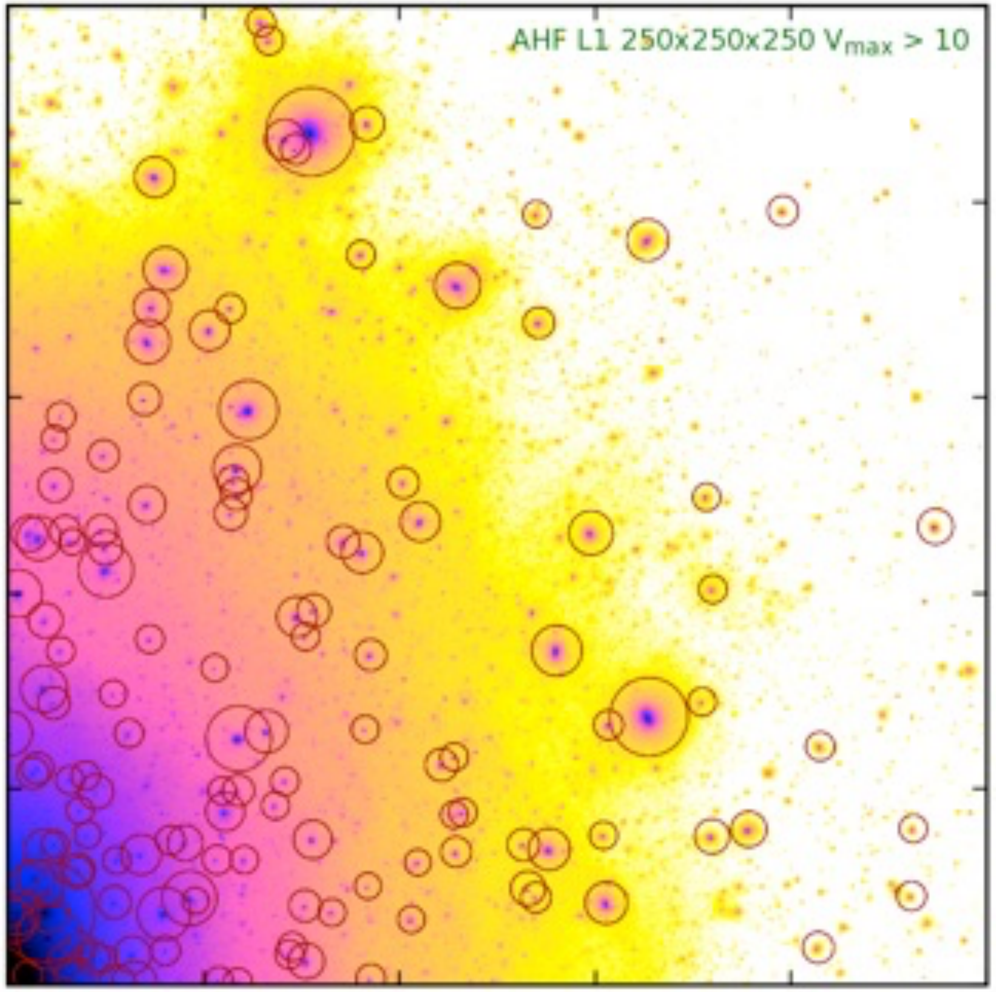
How do we find structures?

- We use a 6D friends-of-friends finder (6D FoF)
- This structure finder explicitly does **not require all particles to be in groups**
- Also run an iterative *substructure* search



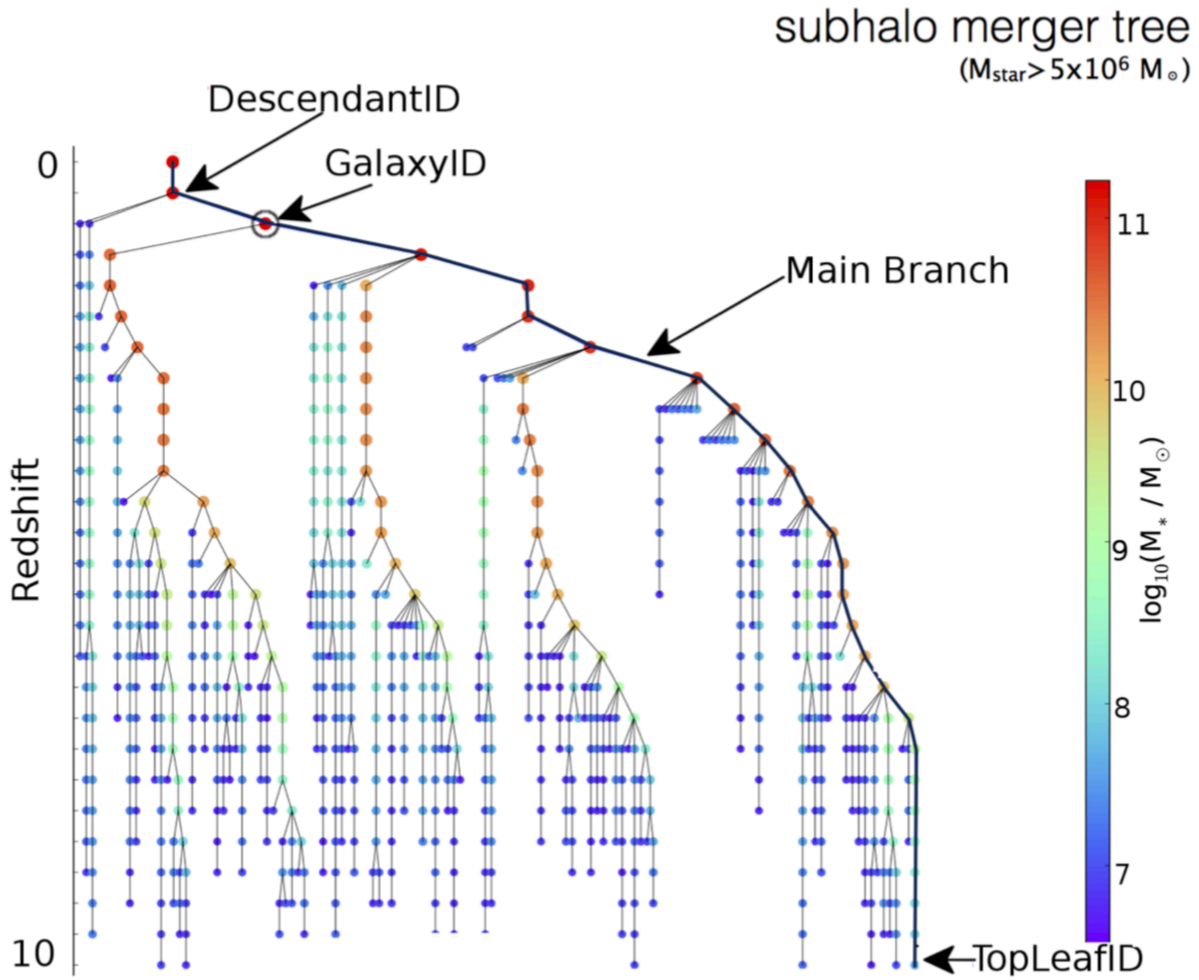
Collecting Metadata

- We run **structure finders on-the-fly** as the simulation is running
- This metadata is saved at higher frequency than snapshots
- Allows us to extract **bound structures from snapshots instantly** (hdf5 is our lord and saviour)



On-the-fly Analysis

- Structure finder also computes galaxy *properties* at higher resolution than snapshots
- Need to link galaxy data together between snapshots
- Tools like tangos help us further process this data (built on sqlalchemy)



Takeaways

- (Hopefully): cosmological simulations are cool!
- Small gains add up; keep making iterative improvements!
- Running on-the-fly analysis is hugely important
- Metadata saves the day (again)
- Some cool tools: yt, unyt, friends-of-friends for clustering

@JBorrow

joshua.borrow@durham.ac.uk